CLAIMS

1. Method of rendering document data compliant with an XML-based mark-up language, comprising the steps of:

- fetching the document data;
- parsing the document data into a document object model (DOM) representation so as to provide a tree structure, comprising nodes representative of the document data elements including tags and/or attributes;

characterized by

- reconstructing the document object model (DOM) representation into a reconstructed document object model (DOM) representation by replacing the nodes of pre-specified elements of said document data elements by one or more nodes comprising standard XML compliant elements having standard tags and attributes so as to functionally extend said XML-compliant mark-up language;
- rendering the document data with the reconstructed document object model (DOM) representation.

2. Method according to claim 1, wherein the pre-specified elements are elements with standard tags and/or attributes providing a functionality, the pre-specified elements being replaced by standard XML compliant elements having one or more different tags and/or attributes providing a modified functionality.

3. Method according to claim 1 or 2, wherein the pre-specified elements are elements with custom tags and/or custom attributes, the pre-specified elements being replaced by standard XML compliant elements having standard tags and/or attributes.

4. Method according to any of the preceding claims, comprising the steps of:

- reconstructing the document object model (DOM) representation by replacing a subset of the pre-specified elements of said document data elements by one or more nodes having standard XML compliant elements with standard tags and attributes;
- rendering the document data with the reconstructed document object model (DOM); and
- only upon triggering reconstructing the document object model (DOM) representation by replacing the remaining pre-specified elements of said document data elements by one or more nodes comprising standard XML compliant elements with standard tags and attributes.

5. Method according to claim 4, wherein triggering is caused during run-time by user action or an external event.

6. Method according to any of the preceding claims, wherein:

- the parsing step comprises parsing the document data into a document object model (DOM) representation so as to provide a tree structure, comprising one or more nodes representative of standard XML compliant elements with standard tags and/or attributes and one

or more nodes representative of custom elements with one or more custom tags and/or one or more custom attributes; and

- the reconstructing step comprising reconstructing the document object model (DOM) representation by replacing the nodes of custom elements by one or more nodes comprising standard elements.

7. Method according to any of the preceding claims, wherein the step of reconstructing the document object model (DOM) representation comprises accessing and modifying the DOM representation data by executing program code, preferably script code.

8. Method according to claim 7, wherein a scripting language, for instance Javascript and/or VBScript, interpreter is applied to execute the script code.

9. Method according to any of the preceding claims, wherein the step of reconstructing the DOM representation comprises the steps of:

a) traversing the DOM representation node for node recursively;

b) upon detecting a node that is to be replaced:

b1) creating a new node of standard elements;

b2) optionally creating a subtree of one or more additional nodes by adding these additional nodes to the newly created node;

b3) inserting the new node and the additional nodes, if any, into the parents

children list of the document object model (DOM) representation on a position immediately before or after the node representative of the node that is to be replaced;

b4) removing the node that is to be replaced from the document object model (DOM) representation;

b5) moving one or more children of the removed node that was to be replaced to the new node or to a preset additional node, if any, that is part of the subtree the new node is root of.

10. Method according to claim 9, comprising the step of mutually connecting the new node and said node that was to be replaced .

11. Method according to claim 10, wherein the step of connecting comprises providing both the node of the node that was to be replaced and the new node with an attribute containing a reference to one another.

12. Method according to any of the preceding claims, comprising after the step of parsing the document data into a document object model (DOM) representation the additional step of saving the current document object model (DOM) representation in a document object model (DOM) shadow representation.

13. Method according to claim 12, comprising saving the DOM element attributes first child, last child, next sibling, previous sibling, and parent node.

14. Method according to claim 12 or 13, comprising traversing the DOM representation node for node and saving the DOM representation into the DOM shadow representation, by

a) starting with some root node;

b) initializing at least the attributes representative of the first child node, the last child node, the next sibling node, the previous sibling node, and the parent node with predefined start values, preferably null values;

c) detecting if the node has a child node;

d) if the node has a child node, then add that child node to the node in the DOM shadow representation by updating the values of attributes firstChild and lastChild of the node and updating the values of attributes previousSibling, nextSibling and parentNode of the child node and, where necessary, its new siblings;

e) repeating steps b-d for every further child node.

15. Method according to any of the preceding claims, wherein the step of fetching the document data comprises fetching the data from a remote server.

16. Method according to any of the preceding claims, wherein the step of rendering the document data with the reconstructed document object model (DOM) representation is performed by a standard webbrowser, preferably Microsoft Internet Explorer 5.0, 5.5, 6.0 or higher, Mozilla 1.1 or higher, Netscape Communicator 7 or higher, Opera 7 or higher or Safari

1.1 or higher, including any webbrowser applications based on the technology of these standard browsers, or so-called 'derived browsers'.

17. Method according to any of the preceding claims, comprising:

- reconstructing the document object model (DOM) representation by replacing at least one node of a pre-specified element of said document data elements by one or more nodes with intermediate custom elements,

- rendering the document data with the reconstructed document object model (DOM) using the at least one intermediate custom element; and

- upon triggering the step of reconstructing the document object model (DOM) representation by replacing of the at least one node of the at least one intermediate custom element by one or more nodes comprising standard elements having standard tags and attributes.

18. Method according to any of the preceding claims, wherein the XML compliant document data is an XHTML- or HTML document or a document with a syntax that complies to any language derived from XHTML or HTML.

19. Method according to any of the preceding claims, comprising the step of dynamically adding one or more new elements to an existing element.

20. Method according to any of the preceding claims, comprising the step of dynamically replacing

one or more existing elements by one or more new elements.

21. Device of rendering document data compliant with an extended XML-based mark-up language, the document data being stored on a remote server and accessible through a network, the device comprising:

- an interface for retrieving the XML compliant document data from the server;

- a parser for parsing the document data into a document object model (DOM) representation so as to provide a tree structure, comprising nodes representative of the document data elements including tags and/or attributes;

- a reconstructor for reconstructing the document object model (DOM) representation into a reconstructed document object model (DOM) representation by replacing the nodes of pre-specified elements of said document data elements by one or more nodes comprising standard XML compliant elements having standard tags and attributes;

- a renderer for rendering the document data with the reconstructed document object model (DOM) representation.

22. Device according to claim 21, wherein the pre-specified elements are elements with standard tags and/or attributes providing a functionality and the reconstructor is adapted so as to replace the pre-specified elements by standard XML compliant elements having one or more different tags and/or attributes providing a modified functionality.

--

23. Device according to claim 21 or 22, wherein the pre-specified elements are elements with custom tags and/or custom attributes and the reconstructor is adapted so as to replace the pre-specified elements by standard XML compliant elements having standard tags and/or attributes.

24. Device according to claim 21, 22 or 23, wherein the reconstructor is adapted so as to perform the method steps of one of the claims 4-20.

25. A system for rendering XML compliant document data, comprising a host computer on which the XML compliant document data are stored, a client computer, and a network connecting the host computer and client computer, wherein the client computer comprises:

- a network interface for retrieving the XML compliant document data from the host computer;
- a parser for parsing the retrieved document data into an object model (DOM) representation so as to provide a tree structure, comprising nodes representative of the document data elements including tags and/or attributes;
- a reconstructor for reconstructing the document object model (DOM) representation into a reconstructed document object model (DOM) representation by replacing the nodes of pre-specified elements of said document data elements by one or more nodes comprising standard XML compliant elements having standard tags and attributes;

- a renderer for displaying the document data with the reconstructed document object model (DOM) representation.

26. System according to claim 25, wherein the reconstructor and/or renderer are adapted so as to perform the method steps according to one of the claims 1-20.

27. Device comprising a computer program product and at least one processor, the computer program product comprising instructions for causing the processor to execute the method steps of any one of claims 1-20.

28. Data carrier containing a recorded computer program product upon whose execution by a processor the method according to any of claims 1-20 is carried out.

29. Computer program for carrying out, when run on a computer, the steps of any of the method claims 1-20.